

SAS® Enterprise Guide® - Your Gateway to SAS®

Dave Shea, Skylark Limited, Wellington, New Zealand

ABSTRACT

SAS® Enterprise Guide® has become the place through which many SAS users access the power of the SAS® System. Some like it, some loathe it, some have never known anything else. Enterprise Guide offers a point and click environment to constructing your code, but also allows the semi-colonista to have their place in the sun too.

INTRODUCTION

This paper aims to show the user some useful and perhaps surprising features of SAS® Enterprise Guide®, but it is not trying to be a comprehensive manual. I aim to focus on areas that my colleagues and clients have found to be useful. This paper was written with Enterprise Guide version 5.100 in mind.

Enterprise Guide provides a point and click access to the SAS® System, but more than that, I believe that it offers a place where much of the SAS System's features are exposed to the user in an easily understandable and accessible way.

Readers might like to download the accompanying Enterprise Guide project and Excel workbook package so that they can experiment with what is discussed with working detailed examples.

The example package may be downloaded via the author's sasCommunity.org user-page:
http://www.sascommunity.org/wiki/User:Downunder_Dave

ENTERPRISE GUIDE – IT'S IN THE WAY THAT YOU USE IT

I make no bones about it, I like SAS and I like Enterprise Guide.

Enterprise Guide is a very nice place to work with SAS. It is not the be all and end all, and I know that it has its critics, but as a way of exposing the power of the SAS System, it's pretty good.

Enterprise Guide allows a user to do their work, explore, learn, or just play. It's all in the way you use it. The point and click environment works well for many people. The whole point and click metaphor has opened up the processing power of a silicon chip for billions of people world-wide. Without point and click, where would you be today ? Not everyone gets excited by a semi-colon.

Enterprise Guide has encouraged and enabled me to explore and feel competent in areas of SAS that I have always wanted get to grips with, but either the learning curve seemed too steep or my time was too short, or both. I would never have tried to work with HTML within my SAS jobs, nor would I feel much happier with Tabulate and Graph were it not for Enterprise Guide.

ENTERPRISE GUIDE – IT IS STILL SAS UNDER THE HOOD

It is important to remember that despite all the icons, dialog boxes and its point and click, when the Run button is clicked, it is still a real SAS session that is fired up and runs your code. That's SAS; warts and all, with its semi-colons, its proc steps, its DATA steps and its options, just SAS.

Enterprise Guide offers an alternative way to running SAS code and also an alternative way of writing or, more accurately, constructing your SAS code.

Enterprise Guide has a bit of a secret that it is not very good at keeping; it is having an affair with PROC SQL and some SAS users feel jealous and let down. Almost all of the SAS code that is generated by tasks in Enterprise Guide makes use of PROC SQL as a surrogate DATA step. For SAS users who are not experienced in SQL or who do not feel the urge to dip their toe into the water, this is one of those stumbling blocks that prevent them getting on with Enterprise Guide.

If you are new or unfamiliar with Enterprise Guide, the glossary at the end of this paper should be useful to you, it is strongly suggested that you go now and familiarise yourself with the terms (see page 12).

ENTERPRISE GUIDE – TWO CAMPS AND THREE GROUPS

It is the point and click method of construction of your SAS code which divides the users into broad two camps: "I love

it!" and "Doh!" This leads me to define my different groups of Enterprise Guide users.

I believe Enterprise Guide users can be put into three main groups, I do not consider one as superior to any other, each have their own needs.

GROUP 1 - I DON'T KNOW WHAT SAS IS, BUT I CAN USE A MOUSE AND I KNOW WHAT MY BUSINESS NEEDS ARE

These people usually take to Enterprise Guide like a duck to water. They do not initially care about what is going on behind the scenes, they have a job at hand, and they can drive Enterprise Guide with their mouse and a little bit of keyboard. The Process Flow shows them what thing produces what output.

The Process Flow also allows them to see the big picture, step-by-step, what happens first, what happens next and so on.

Most Group 1's love Enterprise Guide from day one. They do not really care about Group 2 and Group 3 people, they can just do their job and they are happy.

GROUP 2- I'VE USED SAS BEFORE, BUT NOW MY COMPANY HAVE MOVED TO ENTERPRISE GUIDE AND I LOVE IT!

These people have used SAS in their current or previous roles but never really felt that they understood what was going on. Almost all of the time, their SAS code ran, perhaps after a few false starts, but they were happy. They used some parts of SAS that they weren't all that sure about, but the numbers looked just right. They sometimes used whole chunks of SAS code that a Group 3 person gave to them with a cheery, "don't worry about how it works, just run it, trust me....."

People from Group 2 would like to be Group 3 people, but they really just never have the time to understand all that stuff. They have probably never seen the SAS Language Reference, let alone started reading it from chapter 1. There are still some areas of SAS that leave them exasperated, "what is an INFORMAT ?,and that PROC TABULATE, well, don't get me started....." and sometimes, they end up just "doing it in Excel".

Group 2 people like Enterprise Guide because it lets them do their job and they can see how their SAS processing works in a cleaner way. It also allows them to play with the outside world without too much fuss, bringing their data in from Excel, exploring it, charting it, analysing it, adding value to it and maybe, even then, sending it back to Excel.

GROUP 3 - I 'VE USED SAS BEFORE, BUT NOW MY COMPANY HAVE DONE SOMETHING REALLY STUPID.....

These people have used SAS longer than anyone, always in Display Manager®, some even in line-mode under MVS, "*before that wimpy Display Manager came along and suddenly everyone was a SAS programmer!*"

The Group 3 people saw Enterprise Guide coming a long way before it arrived on their desktop. They conferred, loudly, amongst themselves, each as a latter day Paul Revere, "Enterprise Guide is coming!". According to the Group 3's, it was a BAD THING from the start and would only lead to trouble. Once Enterprise Guide was installed on their desktops they only ever saw problems. Sometimes these complaints were justified, but alongside them, the Group 1 and Group 2 folk were still doing their thing quite happily.

ENTERPRISE GUIDE AS A BUSINESS TOOL

Enterprise Guide is a great place to start using SAS from the ground up. Like Excel, it starts with a blank canvas and offers a small set of main menu items to start off from. Context menus offer close-up assistance.

n.b Whilst these context menus are often referred to as "*right-click*" menus, some Southpaws swap their mouse-keys around, hence this paper will use the generic term "*context menu*".

Importing data from the outside world has often been a stumbling block for new and experienced SAS users alike. Each version of Enterprise Guide has provided an easier and neater way of getting your data into a SAS dataset, from which to start doing your real work.

If you plan to explore and analyse some data and then produce a report on that data, but you do not want or have time to learn the SAS language, Enterprise Guide is certainly your first point of call. For an organisation that has recently adopted SAS as one of its tools, Enterprise Guide allows the existing skill sets of its analysts to be used to actually explore and analyse the data without a semi-colon being typed.

ENTERPRISE GUIDE AS A SAS LEARNING ENVIRONMENT FOR NEW SAS USERS.

Enterprise Guide is ideal for users who are just starting out with SAS and are really only doing so in order to achieve their business goals. They would probably really rather they got their job done, so that they could go home, rather than feel that, before they started on the business problem at hand, they needed to learn some computer language. Not everyone gets on well with computer languages.

Enterprise Guide allows the new and inexperienced users to start making the most of the power of the SAS System almost completely through a point and click interface.

ENTERPRISE GUIDE AS A SAS LEARNING ENVIRONMENT FOR EXPERIENCED SAS USERS.

It is this aspect of Enterprise Guide that I hope is most contentious, but is picked up by the SAS community.

When we stop learning, we stop living.

No one knows everything about SAS and no one really knows all that they want to know about SAS. Experienced SAS users have much to offer back to their SAS community, but they also need to keep themselves fresh and up to date. If they are like me, they are acutely aware that there are whole chunks of SAS knowledge that could do with a bit of revision, or even discovering for the first time.

Most importantly, SAS is a living organism. What I learned about the CHARACTER functions in the 1990's has, by now, been heavily augmented. I need to keep up to date.

I use what I know, but I know that there is probably a better way to do it now. The temptation to just keep doing things the way I used to do them is in direct proportion to the lack of time I have to learn new things.

Time is an expensive commodity. None of us would claim to have enough time in which to do all the things that they want to do. Enterprise Guide offers experienced users a quick start to aspects of SAS that they have just never found the time to get to know.

Enterprise Guide makes it easy to learn new or simply, unfamiliar, parts of SAS. I completely sympathise with the "code is good" folk, I love coding. Enterprise Guide is not taking away the ability of the user to hand write code, it is offering the choice of how the user constructs their code. Task nodes allow the user to step through the building of a SAS procedure and, in many cases, to watch the real SAS code being generated as they complete each step of the Task.

Later we will look at SAS® Graph® as an example of an area of SAS which, for me, was ignored and wasted for many years until it occurred to me that Enterprise Guide would provide an easy way to get started and to explore some of the facets of this feature stuffed tool.

WIZARDS AND ADVANCED VIEW TASKS

Some Tasks come in two flavours, Wizard and Advanced View.

The Wizard view of a task takes the complete novice, and with very few steps, produces an acceptable result. The Bar Chart Wizard offers just four steps to get the user from a SAS dataset to a chart and two of those are "*Where is your dataset?*" and "*What do you want for Titles and Footnotes?*" The remaining two steps do the hard lifting to actually construct the code that makes the chart.

Many users of the Wizard are happy to stop there and do not require the Advanced view. However, the Advanced view provides access to a good deal more of the rich features of the PROC GCHART without the user having to know any of the syntax.

It is this aspect of the Tasks that opens much of SAS to all levels of users. Coding something basic in PROC GCHART, such as the type and positioning of a chart's legend can be a fairly mystifying task when done with code. With Enterprise Guide's Bar Chart Task, it is a case of clicking a check box and selecting an entry from a drop-down list.

ENTERPRISE GUIDE – I'LL DO IT MY WAY

I have worked in a number of organisations where experienced SAS programmers have been made to swap Display Manager for Enterprise Guide. Whilst initially upset, these people then realised that, barring a few features, they can carry on writing their SAS code with proc steps and DATA steps, options and semi-colons as if nothing had really happened. They could create new programs and store them outside of the Enterprise Guide environment; they could call previously created SAS programs unfettered from Enterprise Guide. They could, in fact, carry on much as before,

without having to admit to their friends at the local SAS User Group that they were using Enterprise Guide.

However, this attitude probably means that the areas of SAS which they have never got around to investigating will remain, forever, an unknown. Areas that they have never managed to find the time to get into, to learn, will remain unexplored.

Ladies and gentlemen, I was one of these people!

I managed to fight pretty hard to stay away from Enterprise Guide. I spent a lot of time pointing out things that were not right or missing and justifying why I should just stay as a Display Manager man.

Here is a nice quote:

Life is like riding a bicycle. In order to keep your balance, you must keep moving. - *Albert Einstein*

MY SAS NEMESSES, TABULATE, GRAPH AND ODS.

I have been using SAS in every job that I have had for over 25-years and almost all of that was in Base SAS. During that time I have fought many battles with, first TABULATE, then SAS Graph and recently with the Output Delivery System®, ODS. I have read many guides, papers and books by users on these areas. However, I have never felt that I have managed to really get to grips with these areas of SAS and sometimes, SAS and I agreed to disagree and I “JUST DID IT IN EXCEL !”

Enterprise Guide really has provided me with a place to face these three foes, face on, and for us to become, if not friends, then worthy and trusty allies. How did Enterprise Guide achieve this? The point and click interface, the Code Preview window and more lately for me, the Tooltip interface to Help all helped.

PROC TABULATE – THE ENTERPRISE GUIDE WAY

PROC TABULATE and I have had a difficult relationship over the years. When I first saw TABULATE, it was love at first sight. I wanted it for every job I could think of, it made PROC PRINT look a bit weak and pasty. However, TABULATE was very needy and could sulk quite easily, especially if I asked for percentages. Its heritage, coming from the Bureau of Labor Statistics’ Table Producing Language, provides a less than standard SAS syntax when things get a bit rough. When PROC REPORT came along in the 1990’s I thought it was going to be the end of my struggles with PROC TABULATE, I had found a new best friend. But, to be honest, PROC REPORT was a bit of a struggle too, it offered much, in fact, too much, but I became disillusioned. Every now and then I would call up PROC TABULATE and take it out for a spin, nothing heavy, nothing we couldn’t both agree on, certainly NO PERCENTAGES or at least, if I needed percentages, I calculated them before PROC TABULATE.

When I first saw Enterprise Guide and sat down with my “go on, show me what you can do” look, I thought I had found a massive reason to go no further. They had forgotten to include PROC TABULATE! I couldn’t find it, “*Ha! I was right, this Enterprise Guide thing is hopeless, it can’t even handle that simple PROC TABULATE.*”

Of course PROC TABULATE was there, like most tasks in Enterprise Guide, its task name was a bit more descriptive, the Summary Tables task. When I first started using the tasks, it was a bit of a struggle, but with a bit of practice and a few version enhancements, it has become a very useful and simple tool to use. Along with the ability to format the table with fonts and colours and the SAS ability to push output to lots of different destinations, including HTML, the power of PROC TABULATE was increasing. We were becoming closer than ever before.

An example of something that *should* be simple but always felt like hard work was to produce a nested table without each nested column name and its statistic being repeated over and over.

In the Summary Tables Wizard, set the **Analysis variable labels** and **Statistics label** options to **Hidden** in the dialog box to generate this line of code that will be submitted:

```
Financial_Year *(SaleAmount=' ' * Sum=' ' )
```

This will prevent the column label and analysis statistic name being repeated across the columns

In the Advanced View of the Summary Tables Task dialog box, the context menu over the Preview panel allows the various elements of the table to be re-defined from their default display. Now here we can assign our own label or none at all. We can also choose our own label for the displayed statistic, perhaps substituting **Highest Sale Amount**. in place of the rather dull, **MAX**.

The context menu offers the user the chance to change Properties for the whole Table, the Box Area (top left of the table where the rows and columns meet), column and row headings, as well as data values.

The display font for column names and fonts and formats of display values can be changed to suit your needs.

Using the context menu, the following line of code can be generated to beef up our previous example from the Wizard:

```
Financial_Year * SaleAmount={LABEL="Sales ($)" * Sum={LABEL=" "}}
```

But I really wanted to get a little bit fancy with my table. I wanted to highlight the Total line to make it stick out. Again, the same context menu over the Preview table allows the following code to be generated:

```
ALL={LABEL="Total Sales" STYLE={FONT_WEIGHT=BOLD  
FONT_STYLE=ITALIC}}*{STYLE={FONT_WEIGHT=BOLD FONT_STYLE=ITALIC}}
```

All of this of course, without typing any of the code, and that code is certainly starting to look very busy right now.

Colours can be introduced to your table to highlight certain areas, again by using the context menu. One of the attributes of the Font tab is the Foreground and Background colour. By using these buttons and their colour pickers, this, even scarier code, can be generated without any fuss:

```
ALL={LABEL="Product Group Sub-Total" STYLE={FONT_STYLE=ITALIC FOREGROUND=#000000  
BACKGROUND=#FF8080}}*{STYLE={FOREGROUND=#000000 BACKGROUND=#FF8080}}
```

The above few examples hopefully indicate why there are some big advantages to the code generation offered by Tasks. Let the Task do the grunt work and you can come along and fine up the details later if you need to. The Task also helps the wannabe coder learn as they work.

GRAPH – THE ENTERPRISE GUIDE WAY

SAS Graph is an amazingly rich set of graphical reporting tools which knocks spots off other products that I have come across. However, this richness used to come at a big price. Trying to start using even a simple SAS Graph procedure like GCHART could be an exercise in initial excitement, bemusement and then frustration. When I used to see displays at conferences done by people with a lot of experience at SAS Graph I thought that that sort of stuff was way out of my league.

The printed manuals alone for SAS Graph are many, and each thick enough to stun an Ox. There is so much in it. Where do you start ?

Enterprise Guide offers a simple step-by-step way of getting a graphical report from your SAS data and then offering more and more capability as you feel able to take it on. The commonly used Bar, Pie and Line chart tasks come in the two flavours of Task, Wizard and Advanced. This reflects both their popularity amongst all groups of SAS users and the richness of their features.

The syntax for procedures in Graph is as fiendish as that for TABULATE and, I suspect, forces some new Graph users to give up early in the piece.

A user does not have to be a novice to SAS to find Graph a little too rich. The cost of a superb reporting tool is a syntax that demands a high price. Not only is the syntax rich, the fantastic array of the features attached to each procedure are not always that obvious for users to make use of. The superb "How-To Book for SAS/Graph Software" by Thomas Miron, kept alight the hope that I too could use this product. Then Enterprise Guide came along and, after waving and shouting at me for several years, gave me the chance to start scratching away at Graph.

One feature of charts that has always bugged me is this thing called mid-points.

"Excel doesn't have mid-points", I used to shout.

I had made the effort to have nice proper SAS date values in my dataset, even ones shifted to the 1st of each month and formatted nicely to make life simple. I only wanted one column for each calendar month in my data, nothing fancy. Why then, did my bar chart decide to represent my data with columns that didn't appear to match my data.

Mid-points was the cause, apparently. I honestly did not want to hear about mid-points, I wanted a column in my chart for each month of my data, and I wanted to go home now ! I didn't want mid-points!

Enterprise Guide came to the rescue with the sort of prompting that the Task dialog boxes are ideal for.

The check-box labelled **Specify number of bars** in the Bar Chart Task drew my attention and after playing with a few of the radio-button options and the helpful **Examples** pop-up, I entered a straightforward SAS expression like this:

```
'01Jan2013'd TO '31Dec2013'd BY month
```

and my mid-points worries were at an end. The chart looked like I wanted it to. My daily sales data was represented as one column per month and the Bar Chart Task had generated this option to the VBAR statement:

```
MIDPOINTS="01Jan2013"D TO "31Dec2013"D BY month
```

Without the prompting from the dialog box in the Task, I do not think I would ever have come over to feeling easy about doing straightforward charts in SAS.

ODS - THE OUTPUT DELIVERY SYSTEM

In the old days there was really only simple text as an output destination, mostly to a printer.

ODS provides a huge leap in functionality and flexibility, straight out of the box. However, as soon as I started to scratch at ODS and try to understand what it offered and how it worked, I was side-swiped by a plethora of terms and areas that I just didn't want to go to: tagsets, templates, styles, style over-rides....where on earth did I start ?

I needed a way of starting to understand ODS without it causing me a headache.

I also wanted to play with the big guys, get involved with HTML and push my results out in a way that internet browsers could display them. I wanted fonts, colours, tool-tip flags but most of all....I wanted click-through reports. I wanted a report that would have hot spots that my report consumers could click on to navigate back and forwards through the reports to see more details. I wanted my report consumers to only have to have a browser and not know about the SAS System.

Most of all, I wanted that stuff with minimal effort and I wanted it now!

Enterprise Guide was not critical in overcoming this nemesis, but was essential for encouraging me to get stuck in and do something with HTML and SAS. With Enterprise Guide came an encouragement to start playing with ODS. The SASLOG from any output in a project makes reference to ODS. For the uninitiated, it is not terribly obvious what those SASLOG lines are on about but it is clear that ODS is involved. Changing a Result setting in a node's Properties, from the default of **SAS Report** to **HTML** makes the output go to a different destination and is also reflected in the SASLOG.

I knew a little bit about HTML, that is was really only plain text, and an HTML compatible browser would be able to interpret that text and render nice pages. HTML is only one of a large number of ODS destinations, but it is one that I wanted to conquer, if only a little bit.

I got some basic ideas about HTML from this the <http://html.net> web site, which is well worth a look for the beginner as it assumes no knowledge at all and takes things at a gentle pace. No magic applications need to be downloaded or installed, all you need is a browser and a plain text editor such a Notepad.

ADDING TOOLTIPS TO SAS DATA

A simple DATA step can be used to create a column in a dataset which comes alive in HTML and all created and viewed within SAS. Make sure that the Properties of the SAS Code node containing the following code has its Results property set to **HTML**.

The DATA step that follows includes three SAS statements that go to create the final HTML responsive column in the dataset. The first two assignment statements (`HWRatio=` and `HTMLText=`) are there only to keep the final statement simple.

```
Data Classroom;
Set SASHELP.CLASS;
HWRatio      =Put(Height/Weight, 5.3);
HTMLText     ='"'||Trim(Name)||" has "||" a Height/Weight Ratio of: "||HWRatio||"'>";
StudentName  ="<p title="||Trim(HTMLText)||Trim(Name)||">";
Run;

Proc Print Data=WORK.CLASSROOM;
Var StudentName Sex Height Weight;
Run;
```

When the `WORK.CLASSROOM` dataset was viewed as a normal SAS dataset, the `StudentName` column would look like this for Alfred:

```
<p title='Alfred has a Height/Weight Ratio of: 0.613'>Alfred</p>
```

HTML is a pretty noisy form of communication with lots of odd characters, like, `<A /p <`, but SAS can be used to generate most of the noisy stuff and let you concentrate on the content.

The HTML TITLE attribute within the `<p></p>` tag, used above, consists of two elements, the first element is the text that appears when the tooltip is activated and the second element is the text that appears on the page as a hyperlinked piece of text. This usually [appears underlined and in blue](#).

When the dataset is created and the PRINT procedure results are displayed, the user can move their mouse over the StudentName column and be rewarded with a simple tooltip pop-up conveying some custom information about each Student that we defined. Try it !

MAKING YOUR ENTERPRISE GUIDE LIFE EASIER

There are probably as many tips to using Enterprise Guide in a smarter way as there are users. Sometimes we are thrust in front of SAS Enterprise Guide with no introduction and we make the best of it and get our real job done without being given much time to explore the environment. As a result, we miss quite a bit of the functionality on offer.

LIGHT THE BLUE TOUCH-PAPER AND RUN

New converts to Enterprise Guide often use the “light and run” technique which might lead to misunderstanding as well as missing out on some useful stuff. This technique is often used as a result of the Mathematics of Panic,

e.g. `PANIC_FACTOR = (DEADLINE_IMPORTANCE * NEW_TOOL) / DISTANCE_FROM_BOSS_TO_EMPLOYEE`

The temptation to just “get something run” rather than step back and look at the situation, before getting stuck in, is high. The SASLOG in Enterprise Guide, for example, has lines that you didn’t expect to see and may not be easily understood. Each SASLOG generated by any code in Enterprise Guide starts with this:

```
;*';*";*/;quit;run;
```

which is enough to make anyone run away, in fact, to quit and run. Actually, this is an automatic piece of code which overcomes that time honoured problem of unenclosed quotes, or not putting a final RUN, QUIT statement or semi-colon on a piece of code. It acts as a kind of catch-all, “*Mummy-make-it-all-better*”.

This is then followed by more lines that the user did not ask for. It is hard enough job encouraging users to actually read the SASLOG after running a bit of code, so we need to do something to make it more inviting to read.

Add `OPTIONS LINESIZE=145` to your SAS session and re-run your code. The un-asked for SASLOG lines start to look a bit more useful and comprehensible because they do not break and wrap so much. There are some handy automatic macro variables that are declared and populated for you that you may wish to add to your code so that output tells us where it came from.

Here is a typical set of SASLOG lines:

```
1          ;*';*";*/;quit;run;
2          OPTIONS PAGENO=MIN;
3          %LET SYSLAST=WORK.BREAKSEA_SALES_DETAILS;
4          %LET _CLIENTTASKLABEL='Adjustment to Base Data';
5          %LET _CLIENTPROJECTPATH='C:\Userfile\SASProjects51\SGF 2014 Demo.egp';
6          %LET _CLIENTPROJECTNAME='SGF 2014 Demo 2014.egp';
7          %LET _SASPROGRAMFILE=;
8
9          ODS _ALL_ CLOSE;
10         OPTIONS DEV=ACTIVEX;
11         GOPTIONS XPIXELS=0 YPIXELS=0;
12         FILENAME EGSR TEMP;
13         ODS tagsets.sasreport13(ID=EGSR) FILE=EGSR STYLE=HtmlBlue
13         !
STYLESHEET=(URL="file:///C:/Program%20Files/SASHome/SASEnterpriseGuide/5.1/Styles/Html
Blue.css") NOGTITLE NOGFOOTNOTE
13         ! GPATH=&sasworklocation ENCODING=UTF8 options(rolap="on");
```

Any of these these automatic macro variables can be added to a Title/Footnote statement to help output be self-documenting. When a Summary Tables Task node is given the label “Monthly Sales by Region”, it not only identifies itself clearly on the Process Flow, but also populates the , &_CLIENTTASKLABEL macro variable.

Consequently, we can use the following TITLE statement in the Summary Tables Task node
e.g. `Title "This Table was produced by the &_CLIENTTASKLABEL node.";`

which will resolve to the following footnote text when the Summary Tables Task is run:

```
This Table was produced by the Monthly Sales by Region node.
```

ORGANISE YOUR PROJECT AND PROCESS FLOWS

Use multiple Process Flows where necessary to keep your Project neat and manageable.

A brand new Enterprise Guide Project consists of one Process Flow. As a user gets to work it might quickly become apparent that the number of nodes on the screen might get a bit out of hand. Having more than one Process Flow can make your project easier to manage and can help you separate out parts of your process and thinking. I often create a Process Flow called "Playground" that I use to experiment in prior to implementing an idea in the main Process Flows.

Nodes can be moved from one Process Flow to another with a context menu action.

Name your Process Flows for their purpose

Process Flows get a default name of Process Flow, Process Flow 1, Process Flow 2 etc. none of which are particularly enlightening as to their contents. Each Process Flow can be renamed via the Properties panel of the flow.

Arrange the appearance of individual flows on the Process Flows drop-down menu.

The user can navigate from one Process Flow to another using the drop-down menu on the main tool bar. However, the order that Process Flows appear in the drop-down list do not always suit the user's needs. The order can be changed by viewing the Project Tree and dragging the Process Flow entries, on the tree, up and down to achieve the required order.

KEEPING YOUR PROCESS FLOW TIDY

A Process Flow can quickly become a very busy work area with nodes and connectors all over the place. As such it is worth spending a bit of time to keep it manageable. See the *Handy Options To Use in Enterprise Guide* section at the end of this paper for the menu-paths for these ideas (See Page 13).

Turn off Auto Arrange

The default Auto-arrange of tasks on the Process Flow does not suit me. I like to lay out my Process Flow in my own way. Auto-arrange can be toggled on/off with a context menu click from the white-space on the Process Flow.

Limit the number of Datasets that a node can add to the Process Flow

Some SAS code nodes might easily output 10's or even hundreds of SAS datasets. Each dataset output will appear as an individual icon on the Process Flow. Enterprise Guide has an option that limits the number of output datasets that a given node can add to the Process Flow. The code behind the node still generates the datasets, but Enterprise Guide does not clutter up the Process Flow with numerous dataset icons.

Delete unwanted datasets as you go

If, during processing, a dataset is created and then is no longer required, it can be made to disappear from the Process Flow. Using a bit of hand-coded PROC SQL DELETE or PROC DATASETS within a SAS Code node, can both remove the dataset from its host library, and keep the Process Flow tidy.

TASKS VS PROCS – HOW TO FIND OUT WHICH TASK USES WHICH PROC

The View->Task List menu option allows the user to see which Tasks are available:

Tasks by Category shows all available tasks using their Task name, grouped by the categories of tasks that are found in the main Tasks menu. The Describe category shows the tasks available to the user, such as List Data, Summary Statistics, Summary Tables.

Tasks by Name shows all available tasks, in alphabetical order, by either their Task Name or SAS procedure name. This is ideal for those of us that are used to working by SAS procedure name instead of nice Task names. As an example, the List Data Task is actually going to call good old PROC PRINT and the Summary Tables Task is going to call PROC TABULATE. The best part of the Tasks by Name view is that the user can make the view appear in sorted order by either the Task Name or the SAS procedure name. Some Tasks map to more than one SAS procedure. The

Summary Statistics Task maps to the PROC MEANS, PROC SGPLOT and PROC UNIVARIATE depending on how the Task is configured by the user for that particular purpose. Conversely, some SAS procedures map to multiple tasks. PROC GCHART is the SAS procedure called by each the Bar Chart, Pie Chart and Donut Chart Tasks.

The third Task List view is of **Task Templates**. These are templates of Tasks that a user has previously saved, as a way of storing a particular Task in a pre-configured state, to avoid endlessly filling in the same blank Task with standard information. We will look at this feature in more detail more later.

CONTROLLING YOUR CODE

Default run order of un-connected Tasks

If Tasks and Code nodes are added to a Process Flow and those tasks are not linked together this will not prevent the user from doing what they want to do. However this will be more laborious as each node would need to be run one at a time. This will inevitably lead to mistakes and frustration when multiple nodes need to be run.

With Enterprise Guide, the user can execute a single node, a collection of nodes that are highlighted at Run time (they do not have to be a contiguous selection), a branch of linked nodes, an entire Process Flow or even the entire Project. However, the order in which the nodes are submitted for execution will almost always matter. Trying to run the Summary Statistics task on a dataset, before that dataset has been created is going to lead to tears.

Nodes on a Process Flow will default to executing in left-to-right, top-down order, just like reading a book. If your un-linked Summary Statistics task node happens to be a tiny fraction above the node that creates the dataset it wants to use, problems will occur as this unlinked node may execute before its intended turn.

This is simple to overcome and should be an *always-on* habit for users. Link your nodes in the order that you want them to execute. Once they are linked, use the **Run Branch from <Node-name>** option on the context menu available over the first Task node to be run in the sequence on the Process Flow.

Making Tasks run in your order

An Ordered List allows you to define the order that any set of will run. A Project can have many Ordered Lists. Ordered Lists are really useful if the nodes that you want to run are not necessarily all within the same set of linked nodes or even on the same Process Flow. Weekly and month-end reporting may require a different set of nodes to be executed. Here, the user might create two separate Ordered Lists, one for the weekly and one for the monthly reporting and submit each Ordered List as required. Ordered Lists in a Project are stored on their own special Process Flow initially labelled "Ordered Lists", but this can be changed to suit your needs like any other Process Flow.

USING PROMPTS TO DETERMINE WHAT HAPPENS AT RUN-TIME

Enterprise Guide allows the user to be prompted at run-time as to how to run certain pieces of SAS code within the project.

A typical example of a Prompt might be to present the user with a drop-down list of Staff Names and allow them to choose one or more values. The selected values might then drive a WHERE clause to limit rows in a dataset would be included in a report.

The implementation of Prompts acts as a middle ground between SAS macro variables and users who do not want to get involved in anything starting with an ampersand (the **&** character).

A Prompt can be made from one of twelve data-types, not just character and numeric values, but dates, times, ranges of values and even SAS libraries, datasets, variables or colours.

Some Prompt types get some special treatment, those of type Date are presented as a pop-up calendar date-picker to assist the user. A similar pop-up time-picker is available for Time type Prompts.

Prompts can be set up with a step-by-step dialog box and then they can be used throughout the Project. If required, the user can be prompted at any point in the Project to re-value a given Prompt.

Using Prompts for a Filter in Query Builder – A little trick

There is a little trick to using Prompts in Query Builder which can catch out the novice and experienced user alike. Once a Prompt has been defined in a Project it is normally a straightforward case of assigning the Prompt to a node to make its value available to that node. For example, a user may use a Prompt to indicate to Enterprise Guide which category variable to chart in a Pie Chart Task during that run of the code.

The slight exception is with Query Builder, when the Prompt is used in the Filter tab. The check box labelled

“Generate filter for a prompt value (only applies to prompt types)” will allow values returned by Prompts to be handled correctly. For example, if a Prompt has been defined as being able to return multiple values (e.g “Winter”, “Spring”, “Summer”), this check box ensures that the selected items are handed to the SQL WHERE clause correctly.

CONDITIONAL BRANCHING OF TASKS

Earlier we discussed connecting nodes to create a branch which would allow them to be run in a given order. Conditional branching allows us to expand upon this control.

Adding a Condition to branches of Tasks allows a user to implement a super-layer of control over which nodes or collections of nodes on a Process Flow are executed during a given run of a set of nodes. Each of these branching points can be set up with a Condition dialog box that results in a simple *If <this set of conditions is true> then <do this>, otherwise <do that>*.

The Conditional Branching feature does not lend itself to more than the very simplest of on-the-run decision making, but combined with the Prompting within Enterprise Guide there is room for a user to make a more data-responsive Project.

Base your Condition on values in a dataset, dates and times, prompts, macro variables.

Conditional branching decisions can be based on the first and last values in a dataset, the current date and time or values of Prompts or macro variables. The branching decision is defined or re-defined through a dialog box, so no hand coding is required. Multiple branching conditions can be combined with AND/OR logic.

The Process Flow highlights which nodes are involved in a branching decision by displaying a mini-icon of a flag at top right. A Tooltip on each affected node indicates what branching rules are in place for that node. This allows a quick review of the rules without the user having to open the Condition dialog box.

Following a given run, the icons for each node affected by a branching decision are given a mini-icon in top-right to indicate whether that node passed (a tick) or failed (a cross) the branch test on that most recent run. You should note that when a Project containing Conditional Branching is re-opened, the mini-icons on the affected nodes are re-set to the simple flag, rather than the tick or cross, so that you are ready to run again.

WATCHING THE SAS CODE GET GENERATED AS YOU CONFIGURE THE TASK

Many Tasks provide the ability for users to add snippets of their own custom code within the generated code.

The Preview Code button on a Task dialog allows a separate window to be opened, re-sized and positioned so that as the Task is configured by the user, the corresponding SAS code that is generated, can be viewed.

The Pie Chart Task (PROC GCHART) is a great example of being able to see what actions on the Task dialog causes what SAS code to be generated. This is a really good way of learning how to get the procedure’s syntax to do what you want it to do and experiment with features that you may not even have known about.

The SAS code that Enterprise Guide generates

The SAS code generated by the various tasks can appear to be a bit noisy, oddly laid-out, and includes some unusual macro calls that you were not aware of.

A Task node is not a human coder working away under your keyboard; it’s a generic code generating program and as such it probably does not do things the way you do. I like my code nice and neatly laid out, with *my* comment boxes and *my* indenting style. I take quite a bit of effort to ensure that all my code is just the way I like it to ensure that it not only does the job but can be easily read and managed at a later date. The Task nodes do the code generation work, the result of which does not have to be pretty to look.

CODE TEMPLATES AND TASK TEMPLATES – WHAT IS THE DIFFERENCE ?

Enterprise Guide offers the user the opportunity to use the point and click interface to generate code by configuring the various tasks available. Once the Task has generated the code the user has three options available to them, just use the task to do the job, grab a copy of the code as a stand-alone lump of code and put it into a SAS code node, or use the task as it is configured at present as a template for future tasks of this kind.

Create a Code Template

This is the option for the wannabe coder. When a task is configured using the point and click interface, real SAS code is being generated in the background. The “Add as Code Template” context menu item simply copies the code generated by an instance of a Task and pastes it into a new, stand-alone, SAS Code node on the current Process Flow. From that point on, the code is a wholly independent copy of the original which the user can make their own.

The user can edit and run the code as if they had entered it themselves.

The Add as Code Template menu item is not there to replace a coder, it is there to assist a coder who wants to get a head-start on the job. The Code Template gets you 90% of the way to the code that you want so that can then concentrate on the detailed parts that you want to add by hand.

Create Task Template

This facility allows a user to configure a Task from scratch and then save that Task configuration as a Task Template. This Template can then be used as a starting point for Tasks which need to be similarly configured. This prevents repetitive re-work which saves time as well as potential errors. The user need not start from scratch each time.

It took me a long time to get to understand the Task Template as being one of the more useful aspects of Enterprise Guide. Before I got the hang of this, I was often in the situation whereby I needed several similar tasks to be created within the same project or across projects, and each time, for each instance of the task, I started from scratch.

This laborious repeating of previous pointing and clicking led to cries of “*there has to be a better way!*” as well as mistakes. Samuel Johnson might have said “when a man is tired of London, he is tired of life”, but I say, “when Dave has to do the same mouse clicks over and over again, he gets bored, distracted, and he makes mistakes”.

An example might be a case where several Pie Chart Tasks are required within or between Projects. Each Task shares the same source-data, titles/footnotes, size on screen, colour scheme etc. One instance of the Task can be configured and tested and then, when satisfied, that instance can be used to create a Task Template. For the later required Tasks, the Template can be called from the Task Templates section of the Task List and this presents the user with a pre-configured Task that they only need to add their final touches to.

Of course, there was a better way; I just had not given myself enough time to actually understand what features are available.

Task Template Manager

Once I had got the hang of using Task Templates, the Task Template Manager became a good and trusted friend.

The Task Template Manager allows the user to manage previously created Templates, export them or import them. The Task Template Manager can be found at Tasks->Task Templates-> Task Template Manager.

THE SASLOG AND THE PROJECT LOG

When any node is executed in a Project, a node-level log of that run of that node is created and attached to that node. This log will be over-written each time that node is executed.

One complaint from new, and not so new, Enterprise Guide users, is that this log is useless because it is hard to look through the log of several nodes that have run in sequence. Also, you cannot refer back to a previous run of a given node because each time you run the node, the previous log is over-written. This is where the Project Log comes into its own.

The PROJECT LOG

The Project Log is like the SASLOG from Display Manager, except it is persistent between SAS sessions. Thus the Project Log accumulates log output from each executed node, marking each section of the log with the name of the node responsible and the completion datetime of each node.

In previous versions, the Project Log had to be turned ON for each project, but an application wide option now allows the user to always enable the Project Log for new projects. Refer to the Handy Options section at the end of this paper for some ideas on customising your Enterprise Guide sessions to suit you.

DOCUMENT YOUR DEVELOPMENT AS YOU GO

Assign names and descriptions to Nodes on the Process Flow

Each node on the Process Flow allows you to assign a little piece of descriptive text to document your job. Any text amount which is greater than can be displayed on the icon will be displayed in full by either selecting the icon or hovering mouse over the icon to cause the Tool Tip to appear.

Over-ride default node labels

Always try to make the extra effort of over-riding the default node labels that Enterprise Guide provides. This makes your Process Flow a lot easier to navigate through and to understand. When you create several Task nodes of the same type, the default names, Pie Chart 1, Pie Chart 2, Pie Chart 3 etc. do not help much in distinguishing which one

does what. You can simply edit the node's label to add a more descriptive name.

If you find editing the label of a node icon a bit of a fiddly business, you can achieve the same result by viewing the node's Properties via its context menu and changing the node's label there.

Name and Label the resultant dataset from a Query Builder task

In addition, the Query Builder Task (that invokes PROC SQL) has a feature that initially may be misunderstood.

The Query Builder Task is always creating either a SAS dataset a SAS data view (via an SQL CREATE statement) or a SAS Report (via a simple SQL SELECT statement).

The Query Builder window allows the user to name or re-name the resulting SAS object (i.e. the output SAS dataset) directly from the main window via the text box labelled Output Name. To the left of this box is a text box labelled Query Name. This Query Name box only affects the label on the task on the process flow.

The icon representing the resulting SAS dataset/view/report will have a default label made up as follows:

```
QUERY_FOR_<datasource-name>.
```

It is a really good idea to explicitly label the output SAS object as this will be used to identify the dataset icon on the Process Flow. This can be done via the main Query Builder window, using the Options->Results->Output Data->Label text box.

Use Note nodes to flesh out your thinking and to add clarity

Using the Note node in your Project can really help to document your thinking, convey important information to yourself and others. It can also be a bear-trap.

The Note node is a simple text editor. Notes can be added as stand-alone nodes or linked to other nodes to reinforce the relationship of the information that they contain.

Why can Note nodes be bear-traps too ? Note nodes, like any form of documentation, need to be kept up to date and relevant. An out of date comment can be worse than no comment at all. If a Task node was changed fundamentally and the corresponding comment or comments were not changed as well, anyone reading the comments as a guide to the code node is likely to get a nasty surprise.

Note nodes can be connected to Task or Code nodes to make it more obvious what the Note refers to. Alternatively, a Note node can simply sit unattached on the Process Flow. A Process Flow can support multiple Note nodes.

Use Enterprise Guide automatic macro variables in your Titles and Footnotes

Enterprise Guide creates and populates a number of automatic macro variables which are available to the user for titles and footnotes etc. These can help in self-documenting projects and are especially helpful in allowing a report to point back to the node and project from which it came. The macro variables of particular use are all prefixed:

```
_CLIENT
```

e.g The label of the node that generated the report:

```
_CLIENTTASKLABEL='Sales by Product'
```

or

the name of the host project:

```
_CLIENTPROJECTNAME='SGF 2014 Demo.egp'
```

or

the full path and name of the host Project:

```
_CLIENTPROJECTPATH='C:\Userfile\SASProjects51\SGF 2014 Demo.egp'
```

You will see these macro variables and their values at the top of each submitted bit of code.

n.b. These automatic macro variables are emptied of their values at the end of each node.

GLOSSARY/TERMINOLOGY

PROJECT

This is the main container that Enterprise Guide works with. An Enterprise Guide project can consist of one or more Process Flows, each of which contains one or more Nodes. In Excel the equivalent of the Project would be the Workbook.

PROCESS FLOW

A Project consists of one or more Process Flows. A Process Flow is the container of Nodes that make up the SAS

process or processes that you want to run. In Excel, the equivalent of the Process Flow would be the Worksheet, sitting within a Workbook.

NODE

A node is represented on the Process Flow as an icon. Nodes can be for Tasks, Code Nodes, Notes, Datasets, Views, and output of varying types according to your choices and options, e.g HTML or SAS Report or Rich-Text Format (RTF). Each node offers a context menu of actions dependent upon its type.

TASK

A Task is a dialog box that allows you to construct the code of a SAS procedure, such as TABULATE, MEANS, CONTENTS, GCHART, without actually writing any SAS statements at all. However, there are opportunities for users to insert small bits of code into some Tasks. The Task is represented on the Process Flow as an icon. Opening or modifying the Task causes a dialog box to open to allow the user to configure or re-configure the Task.

CODE NODE

The SAS Code node starts life as a blank SAS editor window and allows SAS statements, such as DATA steps, proc steps and options to be manually typed in and submitted for execution.

It is this Code Node that forms the bridge between the user who only ever wants to write SAS statements and Enterprise Guide. Using the Code Node is not an all or nothing thing, of course. Any Enterprise Guide user can mix and match their use of Tasks and Code Nodes to achieve their desired ends.

CONNECTION

This is a small graphical connecting arrow between two icons on the Process Flow and helps direct Enterprise Guide into executing nodes in the required order. Enterprise Guide will automatically create these connections when one node spawns another node, such as a Query Builder Task that creates a SAS Dataset.

However, the user can connect nodes as they see fit to form a chain of nodes that can then be easily submitted in order without having to submit each node, one at a time.

PROMPTS

Prompts can be defined by the user at a Project level to allow a user to provide input about a specific run of a node or nodes. Prompts are realised in the underlying SAS code as SAS macro variables. The magic of the Prompting framework within Enterprise Guide is that simple and informative dialog boxes can be presented to the user at run time to collect information about that specific run. Once a prompt has been populated it can be used throughout the Project to do things such as act as a WHERE clause, contribute to Titles and Footnotes or even highlight categories in a chart.

HANDY OPTIONS TO USE IN SAS ENTERPRISE GUIDE

Tools->View Open Datasets

This menu option allows you to close a given dataset that the Enterprise Guide session currently holds a lock on. One or all open datasets can be freed with this option. This allows you to then delete or rename any previously locked datasets.

Tools->Options->Results->Results General->Managing Results->Maximum number of output datasets to add to the project.

This option allows the user to manage how many dataset icons get added to the Process Flow, this helps avoid too much clutter on the Process Flow.

Tools-> Options->General->Show Full Details when viewing SAS Libraries in File Dialog Details View

This gives much more info than simply filename, create/modify date and file size in a File Open dialog box. This is like a mini Proc Contents report, especially useful in that it shows the number of rows in each dataset. This is really a query on one of the wonderful DICTIONARY views that are always available to the user.

Tools->Options->Results->Results General->Managing Results->Replace Results

To avoid having to face the quite irritating *"Do you want to replace the results from your previous run?"* message box every time you re-run a node, set the above option to *"Replace without Prompting"*.

Tools->Options->Data->Performance->Close Data Grid after period of inactivity

This option is very handy when you are browsing SAS datasets that are shared with other users or may be required to be updated during the night. When you open a dataset, even just for browsing, Enterprise Guide puts a lock on it so that no one else can use it, including, ironically, yourself, via another SAS or Enterprise Guide session. If an overnight batch-job is set up to update this dataset and a user leaves it open in the data viewer in Enterprise Guide when they head off home, the batch-job will fail.

By setting this option, Enterprise Guide will automatically close the data viewer window if nothing has happened to the project for the specified time.

Tools->Options->Tasks->Tasks General->Display all generated SAS code in Task Output

Use this option to have the Enterprise Guide specific SAS code visible in the code generated by a task.

Tools->Options->Results->Results General->Managing Results->Show Generated Wrapper Code in SASLOG

Use this option to have the LOG lines from the Enterprise Guide specific code, shown in the SASLOG.

Tools->Options->General->Enable Project Log for new Projects

This option means that whenever a new Enterprise Guide project is started, the Project Log is turned ON by default. I think that this is a very handy option to use.

CONCLUSION

SAS Enterprise Guide is now a mature part of the SAS System. It acts as a point-and-click driven tool to construct and run SAS code and then to display the results of that code. The SAS code can be a mixture of traditional hand-coded SAS statements as well as the code generated by the Enterprise Guide Tasks.

Enterprise Guide has been rolled out to many SAS users around the world as their method of accessing the SAS system. For new and experienced users alike, Enterprise Guide can offer many easy pathways into previously unexplored areas of the SAS System.

RECOMMENDED READING

- The Little SAS Book for Enterprise Guide 4.2 - by Susan J. Slaughter and Lora D. Delwiche
- The How-To Book for SAS/GRAPH Software – by Thomas Miron
- <http://html.net> - *Use this site for very simple and free tutorials on HTML, Cascading Style Sheets, PHP and Javascript*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Dave Shea
Skylark Limited
Wellington,
New Zealand

Email: DownUnderGlobalForum@gmail.com

sascommunity.org: http://www.sascommunity.org/wiki/User:Downunder_Dave

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.